TO OPTIMIZE TOKENIZATION BASED ASSOCIATION RULE MINING USING ACO

Dr. Vinay Chopra  
Department of Computer Science and Engineering  
DAV Institute of Engineering &Technology, Jalandhar, India

Komal Thakur  
Department of Computer Science and Engineering  
DAV Institute of Engineering &Technology, Jalandhar, India

Abstract— In the term of Data Mining, Association Rule mining has remarkable role. Association rule mining is a popular mining technique that identifies interesting correlations between database attributes. This research paper gives the detailed introduction to proposed and implemented frequent mining algorithm by tokenization approach based on the Apriori algorithm of association rule mining using the Ant colony Optimization (ACO) to find the maximum fitness value by means of parameters like number of rules generated and the execution time consumed.

Keywords— Association rules, frequent item sets, frequent patterns, tokens.

I. INTRODUCTION

Data Mining: Data mining is a process that discovers the knowledge or hidden patterns from the large databases. Data mining is also known as the core processes of Knowledge Discovery in Databases (KDD). It is a [1] powerful new technology with extreme potential to assist organizations to concentrate on the most important information in their data warehouses.

II. LITERATURE SURVEY

On the basis of the association rule mining and Apriori algorithm, the author YK Rana [2] proposed an improved algorithm based on the Ant Colony Optimization algorithm. They optimize the result generated by Apriori algorithm using Ant colony optimization algorithm. The algorithm improved result produces by Apriori algorithm. In another paper, K.Saravana Kumar [3] surveys the most recent existing association rule mining techniques using Apriori algorithm. The conventional algorithm of association rules discovery proceeds in two steps. All frequent item sets are found in the first step. The frequent item set is the item set that is included in at least minimum support transactions. The association rules with the confidence at least minimum confident are generated in the second step. Heitor S. Lopes [4] proposed an algorithm for data mining called AntMiner (ant-colony-based data miner). The goal of AntMiner is to extract classification rules from data. They compare the performance of AntMiner with CN2, a well-known data mining algorithm for classification, in six public domain data sets. In a further paper, Suhani Nagpal [5] improves the performance of the conventional Apriori algorithm that mines the association rules. The approach is to attain the desired improvement is to create a more efficient new algorithm out of the conventional one by adding the encoding and decoding mechanisms to the later in order to demonstrate the importance of the efficient decoding to high data mining performance and from various experiments it is proved that the logarithmic decoding method is the most efficient among the all methods it can speed up all the required processes. Fernando E. B. Otero [6] proposed a new
sequential covering strategy for ACO classification algorithms to mitigate the problem of rule interaction, where the order of the rules is implicitly encoded as pheromone values and the search is guided by the quality of a candidate list of rules. In a further paper, JaiWeiHan [7] proposed a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns, and develop an efficient FP-tree based mining method. Their performance study shows that the FP-growth method is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm and also faster than some recently reported new frequent-pattern mining methods. Meera Narvekar [8] designed a new technique which mines out all the frequent item sets without the generation of the conditional FP trees. Unlike FP tree it scans the database only once which reduces the time efficiency of the algorithm. It also finds out the frequency of the frequent item sets to find out the desired association rules.

III. PROPOSED NEW MINING ALGORITHM
Association rule mining consist of two step
(i) Finding all frequent patterns: using some mining algorithms like Apriori, FP Growth. Also we proposed newer algorithm for frequent pattern mining.
(ii) Generate strong association rules from the frequent patterns: all frequent pattern rules are checked for minimum support and minimum confidence to generate association rules.

The Bit table in the proposed algorithm achieves good performance by significantly reducing the cost of candidate generation and support counting. However, in situations with a large number of frequent itemsets, long itemsets, or quite low minimum support thresholds, it is costly to handle a huge number of candidate sets.

The main improvement of our algorithm is to optimize the frequent single items and those items co-occurrence with them. The data structure Bitable is also used horizontally and vertically to calculate the token array and count supports, respectively. token array and the corresponding computing method are proposed. By computing the token, those item sets that co-occurrence with representative item can be identified quickly. The frequent item sets, including representative item and having the same support as representative item, can be identified directly by connecting the representative item with all the combinations of items in its subsume token. Thus, the cost for processing this kind of item sets is lowered, and the efficiency is improved.

FLOWCHART FOR PROPOSED MINING ALGORITHM
IV. OPTIMIZE THE RULES USING ACO ALGORITHM TO THE MAXIMUM FITNESS VALUE

Each rule have antecedent (left side) and consequent (right side) parts respectively. We assume consequent part of rule as ant and antecedent part’s first attribute as Food. Each rule has occurrence value means how many times that rule occurring in transaction set. Here we would find those who have same ant and same food. e.g. Suppose we see first rule, second rule and third rule has same food and same ant then among them that rule is best which have highest occurrence value in transaction set means that path is mostly followed by ants and it is shortest path. In this way we obtain various groups which have same food and same ant. The occurrence of each rule is assumed as pheromone value for ant. The each group’s best rule is found based on its occurrence. At the end we get only those rules which are optimized by ACO.

ALGORITHM STEPS

Each rule having antecedent (left) and consequent part (right). Consider left part as food and right as ant.

1) Find the rules having same left and right side then count the occurrence i.e. finds the count for the antecedent from the rest of association rules. Assume this count as Fitness1.

2) Fitness2(k)=conf(k)*log(support(k)*length(k)+1)

Fitness (k) is the fitness value of association rule type k. Confidence (k) is the confidence of association rule type k. Support (k) is the actual support of association rule type k. Length (k) is the length of association rule type k. The larger the support and confidence, the greater the strength of the association, meaning that it is an important association rule.

3) Combine fitness1 and fitness2 and select that rule among all the same rules which is having highest fitness value.

V. COMPARISON TABLES AND GRAPHS

1. Comparison table and graph for the existing Apriori algorithm and FP growth algorithm with new frequent mining algorithm by using the parameter for execution time.

<table>
<thead>
<tr>
<th>Minimum Support</th>
<th>Apriori</th>
<th>FP-Growth</th>
<th>Improved Frequent Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.1]</td>
<td>3795</td>
<td>788</td>
<td>146</td>
</tr>
<tr>
<td>[0.2]</td>
<td>1266</td>
<td>338</td>
<td>59</td>
</tr>
<tr>
<td>[0.3]</td>
<td>801</td>
<td>205</td>
<td>48</td>
</tr>
<tr>
<td>[0.4]</td>
<td>724</td>
<td>154</td>
<td>44</td>
</tr>
<tr>
<td>[0.5]</td>
<td>269</td>
<td>122</td>
<td>45</td>
</tr>
</tbody>
</table>

2. Comparison graph for the number of rules generated before applying ACO and after applying ACO on the new proposed frequent mining algorithm.
VI. REFERENCES


